
Subject: Re: [PATCH] Reduce uidhash lock hold time when lookup succeeds
Posted by [Serge E. Hallyn](#) on Fri, 18 Feb 2011 18:25:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quoting Matt Helsley (matthltc@us.ibm.com):

> When lookup succeeds we don't need the "new" user struct which hasn't
> been linked into the uidhash. So we can immediately drop the lock and
> then free "new" rather than free it with the lock held.

>

> Signed-off-by: Matt Helsley <matthltc@us.ibm.com>
> Cc: David Howells <dhowells@redhat.com>
> Cc: Pavel Emelyanov <xemul@parallels.com>
> Cc: Alexey Dobriyan <adobriyan@gmail.com>
> Cc: "Serge E. Hallyn" <serge@hallyn.com>

Acked-by: Serge E. Hallyn <serge@hallyn.com>

And might I say that the label 'out_unlock' in that function is
sadly named :)

> Cc: containers@lists.linux-foundation.org
> ---
> kernel/user.c | 12 ++++++-----
> 1 files changed, 7 insertions(+), 5 deletions(-)
>
> diff --git a/kernel/user.c b/kernel/user.c
> index 5c598ca..4ea8e58 100644
> --- a/kernel/user.c
> +++ b/kernel/user.c
> @@ -157,16 +157,18 @@ struct user_struct *alloc_uid(struct user_namespace *ns, uid_t uid)
> */
> spin_lock_irq(&uidhash_lock);
> up = uid_hash_find(uid, hashent);
> - if (up) {
> + if (!up) {
> + uid_hash_insert(new, hashent);
> + up = new;
> + }
> + spin_unlock_irq(&uidhash_lock);
> +
> + if (up != new) {
> put_user_ns(ns);
> key_put(new->uid_keyring);
> key_put(new->session_keyring);
> kmem_cache_free(uid_cachep, new);
> - } else {
> - uid_hash_insert(new, hashent);
> - up = new;

```
>    }
> - spin_unlock_irq(&uidhash_lock);
> }
>
> return up;
> --
> 1.6.3.3
>
> _____
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containers
```

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers
